```
        ************************************************************
                                          ***
                ////////      ////   ////  ***  /////////
               IIIIIIIII/     IIII/  IIII/    ***IIIIIIII//
              IIIII/          IIII/  IIII/     IIII/  IIII/
              IIIII/          IIII/  IIII/     IIII/* IIII/
              IIIII/          IIII/  IIII/     IIII/***III/
              IIIII//////  // IIII///IIII/  // IIII///***I/  //
               IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIII*** II/
                                                          ***
        ************************************************************
                          CIRCULO DE USUARIOS DE QL
        ************************************************************
```

Fanzine mensual independiente para usuarios de Sinclair QL y compatibles

----

AÑO 3 NUMERO 19 ABRIL DE 1990

Estamos en el año 10 Después de Sinclair. Toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

Compilación de colaboraciones y distribución: Salvador Merino

Para recibir información sobre cómo recibir y/o colaborar en el fanzine, enviad un sobre franquedo y con vuestra dirección a: Marcos Cruz, Acacias 44, 28023 MADRID.

CONTENIDO

Portada de este número: FERRARI, ROBIN, BICIS....

Con este número de CUQ se incluyen los programas siguientes:

- Ferrari_bas.- Un programa que dibuja un ferrari.
- Z88 FORTH V$ 1.00.- Primer FORTH instalado en el portátil Z88. Documentación en formato Pipedream. Autor: Salvador Merino.
- Verb_tsk.- Un programa turbo compilado para crear un diccionario de verbos. Autor: Dasio Carballeira.

Material preparado para los próximos números:

- Z88 FORTH ATACA DE NUEVO
- LO QUE TODOS QUERIAN SABER SOBRE EL ST/QL EMULATOR Y NADIE SE ATREVIA A PREGUNTAR.
- COMENTARIO THE PAINTER Y FERRER.

EDITORIAL

Cuando miramos hacia atrás, nos asustamos de ver cuanto tiempo ha pasado. Especialmente desde aquel dia que nuestro querido QL dejó de reproducirse. Por aquellos dias, los usuarios de QL y ST Españoles creiamos que los PCs debido a su arquitectura bastante anticuada comparada con el MC 680XX, no tenia ningún futuro y sus dias estaban contados. Por desgracia para nosotros, no habiamos contado con el factor MARCA/PUBLICIDAD/IGNORANCIA.

Han pasado muchos años, el QL se ha mantenido gracias al entusiasmo y dedicación de sus usuarios, pero existia un problema muy grave sin resolver. Una máquina fuera de producción con el paso del tiempo dejaria de existir, y todo el software escrito para ella habria sido hecho en vano.

Hace varios años, muchos seguidores del QL estaban esperando el FUTURA. Desgraciadamente, nunca nació y desesperados muchos de ellos vendieron sus equipos y compraron otro 68000 (especialmente los programadores) o un PC (los profesionales usuarios de programas).

En el momento actual el futuro es bastante brillante. El QL ha demostrado ser un monstruo resistente a pesar de sus defectos de fabricación y de diseño. En Quanta hay varios Clonos que necesitan piezas especificas QL para su construcción en casa. El emulador de QL en el Atari ST ha sido perfeccionado en varias versiones, y la última parece bastante interesante. El emulador de QL en el AMIGA, aunque es muy lento en acceso a disco, por su precio no deja de ser interesante. El THOR XVI, un mito fuera del alcance de muchos bolsillos, compatible QL de nacimiento busca hoy la garantia de tener las puertas abiertas en el futuro, el UNIX (una torre de BABEL). El sistema operativo Minerva una alternativa al QDOS para Clonos de Quanta con MC 68020 (el QDOS no sirve). El SMS2 para Atari ST por Tony Tebby,.....

Hoy en dia nuestro futuro está solucionado. Podremos disfrutar del mundo 68000/QDOS hasta el final de los tiempos Sinclairianos o el IMPERIO PC, pero nada más.

El único 68000 importante que no tiene emulador de QL todavía es el Apple Mac. Lo que pasa con ese cacharro es que se supone que tiene una libreria de programas de una calidad tal (incluso se supone que el PC se queda mamando al lado) que no necesita un emulador QL. El otro problema reside en su interface de disco, pues los únicos programas para leer discos PCs que conozco necesitan una unidad de disco standard instalada en el port SCSI, y eso podria resultar bastante caro.

Hace mucho tiempo que no hablo de los Transputers, no recuerdo si he mencionado alguna vez que pasó con ellos. El pasado mes de febrero Serafín se intereso por el tema y por ese detalle voy a recordar la historia. Leon Heller fundo Tinkerers Association alla por principios de los 80 o finales de los 70. Esa misma asociación se convirtió antes de que el propio QL viese la luz de la calle en lo que hoy conocemos por QUANTA. Lo que significa que habia más de 300 socios sin QL. Aquellos miembros eran usuarios del TRS-80 y el Video Genie (un clono del primero). La cuestión es que el TRANSPUTERS para QL fue una realidad en 1988 con interface, modulos transputers, compiladores y alguna que otra aplicación para demostración. Pero en ese mismo año hubo en Quanta un lio interior, Leon Heller perdió el control y Roy Baber (el nuevo editor que ha dimitido recientemente) censura los articulos de Leon Heller con la excusa de que los Transputers es un mundo que no tiene nada que ver con el QL.

Ahora mismo lo único que sé es que se han usado Transputers en Quanta para construir interfaces de disco duro. Y que igual que han hecho en el mundo PC, hemos censurado al mundo Transputers.

Tio Clive Sinclair se le escapo una vez que estaban diseñando un nuevo ordenador personal muy potente de 32 bits, pero no dijo en que estaba basado, pues era un super secreto en diseño. Pero que estaba muy impresionado por el diseño de Transputer y habia examinado la posibilidad de usarlo.

Hoy estoy contento porque el pasado dia 1 de marzo he enviado una copia de mi Z88 FORTH V$ 1.00 a Z88 EPROM (el club de usuarios Z88 Inglés). Para un Hacker dominguero como yo, es un triunfo haber conseguido tal objetivo, y haber sido el primer programador en instalar el FORTH en el Z88.

En algunos de los últimos CUQs habia gente que hacia mil y un invento para ahorrarse un duro en el cartucho de cinta de impresora. Si bien es cierto que el

cartucho en impresoras como la BMC de INVES cuesta más de 1500 ptas, en mi nueva Epson LQ-550 de 24 agujas cada cartucho nuevo me cuesta en el Corte Inglés 620 ptas.

Me ha informado Luis Valero que ha comprado, él y un amigo, en el Corte Inglés (más publicidad gratis) un cable para conectar el QL a un TV con toma EURO-conector y se consigue una calidad de monitor con toda la pantalla MODO 4 por solamente 1.500 ptas.

Salvador Merino

NUEVA OPCION PARA OBTENER CUQ LOS NO COLABORADORES

Aprovechando la última bajada de precio de los discos 3.5" podemos ofrecer a todos aquellos que no deseen colaborar mensualmente, pero si recibir CUQ todos los meses, una nueva opción que consiste en 4 discos/números CUQ por 1.000 ptas con todos los gastos incluidos.

Si te decides por esta opción aún puedes enviar alguna colaboración en papel si solamente ocupa un folio o en disco, pero no debes enviar ningún sobre con sellos de retorno, pues se te devolvera en el mismo sobre que contiene tu disco CUQ del mes.

Para pedidos de atrasados ofrecemos otra variante, pues tal es el volumen de material almacenado que volverlo a grabar es un trabajillo bastante pesado.

Disco CUQ I contiene: 1-2-3-4
Disco CUQ II contiene: 5-6-7-8
Disco CUQ III contiene: 9-10
Disco CUQ IV contiene: 11-12-13
Disco CUQ V contiene: 14-15
Disco CUQ VI contiene: 16-17
Disco CUQ VII contiene: 18-19

Cada disco CUQ de la lista anterior costaria 300 ptas con todos los gastos incluidos.

Solamente se acepta como forma de pago Giro Postal o efectivo (cheques y tarjetas no son aceptadas). Por ahora, mientras no salgan otros distribuidores alternativos, la dirección es:

Salvador Merino
Ctra Cádiz, Cerámicas Mary
29640 Torreblanca del Sol
Fuengirola (Málaga)
Teléfono (952) 475043

Como es natural, toda esta nueva opción es una pequeña prueba para comprobar si tiene aceptación, pues en caso contrario volvemos y seguimos como siempre. Además yo soy el primero que no está muy seguro si voy a poder llevarlo adelante sin que ello me robe tiempo en otras tareas (p.e.: Martes oración y ensayo de coro, Miercoles lección de Biblia, Viernes reunión de mi grupo, sábado reunión general y festejo de cumpleaños y otros, Domingo ensayo de coro y actuación en vivo. Y en mi trabajo no puedo usar el QL solamente el Z88). Y también hay que agregar que, si os molestais en hacer un poco de matemáticas, he calculado los costos tan bajos que si el precio de los discos subiese o correos le da por cambiar las tarifas, habriamos liado la de San Quintín o la chapuza del siglo.

Si no es mucha molestia es obligatorio rellenar la siguiente ficha todos aquellos que opten por esta nueva opción (para los colaboradores no es muy necesario, pero casi):

C.U.Q.
======

APELLIDOS..NOMBRE.......................
DIRECCION..LOCALIDAD,,,................
TRABAJO..TELEFONO PARTICULAR..........
DIRECCION..TELEFONO TRABAJO.............

EQUIPO:
QL BASICO VERSION..........................

AMPLIACION MEMORIA MARCA..........DE .....K
DISCOS....................................
CONTROLADOR...............................
IMPRESORA.................................
OTROS.....................................
....

CAMPOS DE INTERES...
....

NOTICIAS

EL EMULADOR DE QL EN EL AMIGA YA DISPONIBLE AL PUBLICO

El nombre del emulador de QL en el Amiga es KOWALLIK emulator o "Quantum Leap Emulator". Se encuentra disponible por 3 (Catalogue code 'USPEC 13') en:

SOFTVILLE PD/SHAREWARE SERVICES
Unit 5, Stratfield Park, Waterlooville, Hampshire, PO7 7NX. (UK)

La forma de pago normalmente es Giro internacional de 5 (2 de gastos de envio y embalaje).

El autor del emulador es:

Rainer Kowallik, Eisackstr. 14, 1000 Berlin 62, West Germany.

El emulador corre los programas QL al doble de la velocidad normal de un QL. Para la emulación de la pantalla se ha usado el chip BLITTER conseguiendo una actualización de la pantalla muy rápida.

Por lo visto, las instrucciones 'TAS' (Test And Set) usadas muy frecuente en el QL fallan en el AMIGA por un defecto de hardware. Rainer ha solucionado el problema escribiendo un programa que reemblaza cualquier ocurrencia 'TAST' en el software QL con el código apropiado para que el Amiga se lo pueda tragar.

El mayor problema de la versión disponible en Noviembre'89 es un bug en la rutina de escritura de discos. Un disco escrito por el QL puede ser leido en el Amiga en modo emulador, pero un disco escrito por el Amiga no puede ser leido por un QL. El problema se debe a su rutina de calculación 'CRC' en el código floppy-driver. El autor dice que el problema será solucionado muy rapidamente. Hay planes para instalar el driver NETWORK y la RAMDISK (disponible en el Amiga en modo nativo). También se desea agregar los 32 colores AMIGA (los modos gráficos nativos del AMIGA).

El programa viene instalado para correr en un teclado AMIGA Alemán. Para adaptarlo a otro idioma, hay que alterar las tablas de códigos en el listado assembler del emulador y volverlo a ensamblar (con el emulador se entrega todo el programa FUENTE), o utilizar la utilidad ALTKEY del TOOLKIT II.

Exceptuando los MICRODRIVES y la NETWORK, todos los dispositivos Standard están implementados: CON_, SCR_, PIPE_, SER1, PAR, FLPn_.

El principal defecto es el acceso a disco, el cual es muy lento comparado con un QL. ¡Tarda 8 minutos en formatear un disco!. La lectura es el doble de rápida que la escritura (pero es 1/4 de la velocidad de un QL).

Hay también rutinas para transferir ficheros via serial link entre máquinas (Amiga/QL link).

THOR INTERNACIONAL BUSCA PROGRAMADORES

THOR INTERNACIONAL busca 20 programadores assembler 68000/8 con mucha experiencia para participar en el desarrollo del sistema operativo UNIX y programas de aplicaciones. Enviar una carta o un FAX y curriculum a: THOR INTERNACIONAL, PO Box 59, DK 1002 Copenhagen K, Demmark. FAX No. +451 938292.

NUEVA VERSION QL EMULATOR EN EL ATARI ST

Ahora soporta HyperCache, una rápida tarjeta procesador para ST, la cual

duplica la enorme velocidad del ST. También se venden máquinas instaladas con teclados Inglés, Alemán y Francés y TOS.

ATARI MEGA ST1 749 libras. ST2 899 libras. ST4 1199 libras. Incluido p&p, pero sin monitor.

Salvador Merino, 22/2/1990

MAQUINA DE TRADUCCION

El 25 de enero de 1988 la empresa holandesa de programación BSO hizo una demostración a la prensa esperantista del prototipo de "DLT" (Distribuita Lingvo-Tradukado), un sistema de traducción automática que desarrolla desde 1982.

DLT fue proyectado para traducir textos comerciales, técnicos e informativos, de y hacia diversas lenguas en redes de ordenadores. Se espera la primera versión comercializada para 1993, según el pronóstico actual. La primera versión traducirá del inglés al francés; se le añadirán otras lenguas durante los años 90.

La construcción de sistemas de traducción automática es actualmente un campo tecnólogico en rápido crecimiento en todo el mundo, como quedó claro en la rueda de prensa en Utrecht.

El programa DLT destaca por el peculiar y único sistema adoptado por las personas que lo han desarrollado en BSO. Puede calificarse mejor como una red de estaciones de traducción interactuantes que como un único sistema de traducción. Deben imaginarse tales estaciones como ordenadores personales, en el marco de una oficina del futuro o integrados con la electrotecnia del hogar de los años 90. Cada estación en la red se ocupará exclusivamente de la traducción a o desde la lengua elegida por el usuario.

El nexo común en cualquier red de traducción DLT se denomina "interlengua", la cual se mantiene invisible y desapercibida para el usuario, pero en la cual una versión compacta y normalizada del contenido del texto es distribuida a las estaciones receptoras. BSO eligió para este papel de interlengua al Esperanto.

El prototipo mostrado en la rueda de prensa es todavía sólo una versión de laboratorio, con un modesto diccionario de aproximadamente 2000 palabras inglesas, pero contiene los algoritmos básicos del futuro modelo. Las frases tecleadas son primeramente traducidas al Esperanto. Si hay más de un posible traducción, las interpretaciones más probables son separadas de las más improbables por medio de una búsqueda en un "banco de conocimientos", que consiste en aproximadamente 80000 "unidades semánticas". Para conseguir esto, los programas de BSO emplean métodos elaborados por medio de investigaciones en Inteligencia Artificial. Después de que el usuario confirma que la interpretación elegida es la adecuada, el programa traduce el resultado al francés.

El empleo de métodos de Inteligencia Artificial en las máquinas actuales ralentiza un poco el proceso de traducción. BSO espera que este problema desaparezca cuando la próxima generación de ordenadores, los denominados ordenadores de proceso paralelo, esté disponible a comienzos de los años 90. En la situación actual es más importante que los programadores puedan estudiar en detalle la anatomía del sistema. El proceso de traducción puede ser contemplado por medio de cinco terminales gráficas como una película a cámara lenta, en la cual las frases a traducir en cada etapa aparecen en la forma de estructuras arborescentes.

Aunque el proyecto DLT ha consumido ya la mitad del tiempo asignado para exploración y desarrollo, todavía falta mucho trabajo para construir un sistema completo. Eso incluye la ampliación de los diccionarios y los bancos de conocimiento. Fue dado a conocer en la rueda de prensa que un tercio, aproximadamente 6 millones de florines, de los recursos disponibles para el periodo 1982-1991 ha sido ya empleado. Esto quiere decir que el proyecto queda estrictamente en los límites de su presupuesto. Del coste total, calculado en 18 millones de florines, el 50% ha sido cubierto por la contribución del Ministerio de Economía de Holanda, de los fondos disponibles para investigación en nuevas tecnologías, y la restante mitad es pagada por la misma BSO.

Según la opinión del jefe del proyecto DLT, Toon Witkam, BSO ya se ha colocado en una muy buena posición entre los diez primeros -principalmente japoneses- competidores en la carrera de alta tecnología de los principales proyectos en este campo. Para continuar la evolución y comercializar el sistema DLT, BSO busca ahora un socio entre las grandes empresas de electrónica y ordenadores. Su apoyo será necesario para que la interlengua de DLT se convierta en el estándar industrial en las redes multilengua de ordenadores. Esa estandarización permitirá unir diversos tipos de ordenadores a la red de traducción DLT. También grandes editoriales son consideradas como posibles socios.

Hasta ahora, el proyecto lo lleva a cabo BSO en solitario, sin ninguna otra empresa asociada. También se caracteriza por una gran transparencia informativa y una gran abundancia de contactos internacionales. Lingüistas de diversos países (inclusive Finlandia, Hungría y Japón) ya añadieron su contribución al trabajo de los 12 investigadores en Utrecht. En Holanda, una investigación de apoyo se lleva a cabo en la Universidad de Amsterdam.

(Artículo aparecido en la revista "Kontakto" en abril de 1988, traducido de la lengua internacional Esperanto por:)

Marcos C.M., Madrid, 13/1/1990

CARTAS ABIERTAS

CARTA DE PRESENTACION
SOLICITUD DE INSCRIPCION

Soy Félix Alonso, con domicilio en Burgos, Plaza Francisco Sarmiento, número 2, 6º A-4, de profesión administrativo-contable, usuario de QL casi desde que Investrónica presentó la versión española, y exsocio del Club QLAVE, al que pertenecí hasta finales de 1988.

En la actualidad dispongo de dos máquinas QL,con ampliación a 640K RAM (con SuperQboard), dos unidades de Disco, Monitor Philips monocromo, ratón e Imprsora Epson LX-800.

Por mediación de Salvador Merino he sido informado de la creación y existencia del nuevo Club CUQ, en el que lógicamente me interesa integrarme, por lo que SOLICITO mi inscripción como socio del mismo, en la confianza de que caminando conjuntamente, conseguiremos utilizar más y mejor sus posibilidades, y éllo redundará en una mayor estima y valoración de nuestro querido QL.

Félix Alonso, Burgos,19/2/1990

UN NUEVO LECTOR

Me llamo Roberto y vivo en un pequeño pueblo de la Ribera de Navarra (Fitero), y trabajo en una fábrica de piedra de alabastro que hay aqui. Para sacarme algo para gastos y practicar contabilidad utilizo el ordenador, utilizando los programas que me dieron al comprar el ordenador.
El equipo que tengo está compuesto por un QL ampliado a 640 K, 2 unidades de disco 3.5" y una impresora de carro pequeño Brother.

Roberto Muro Torrecilla, 16/2/1990
C/ Angós - Parte de atrás, 31
31593 FITERO (Navarra)
Tfn.(948) 776463

COMENTARIOS DE LIBROS Y PUBLICACIONES

LISP

Cuando recomendé libros acerca del LISP en el artículo que acompañaba al L-TOOLKIT me dejé un título en el tintero, a lo mejor porque no se trata de un libro recomendado. Éste es "Artificial Intelligence on the Sinclair QL" escrito por Keith y Steven Brain (padre e hijo) y publicado por Sunshine (en España lo distribuye RA-MA). Es el único libro sobre IA escrito específicamente para el QL.

Todos los capítulos contienen ejemplos escritos en SuperBASIC y uno de ellos está especialmente dedicado al ARCHIVE. Trata la IA a un nivel elemental, y no creo que sea precisamente el mejor libro para ir familiarizándose con los conceptos de IA, es más bien un ejemplar para bibliófilos.

Sergio Montoro Ten
Madrid, 23 febrero 1990

SPECTRUM

DISCiPLE

Pasar las pantallas y los programas del SPECTRUM al QL puede hacerse a través de discos o microdrives en lugar de usando la RS-232 o el network. El principal problema de la RS-232 es la dificultad para encontrar un cable apropiado y con respecto al network, para que vamos a hablar, funciona cuando quiere y se producen errores (por lo menos en mi QL JM).

Si se tiene el DISCiPLE pueden escribirse las pantallas en un disco en blanco usando los comandos para escritura directa de sectores:

SAVE @ unidad,pista,sector,direccion_de_inicio (ver pág. 39 del manual)

Con este comando podemos escribir, por ejemplo, 14 sectores consecu- tivos de 512 bytes cada uno; de modo que comenzando en la dirección 16384 (el inicio de la pantalla) hasta la 23296 (el final de los atributos) todo quede grabado secuencialmente. Luego, usando el Super Media Manager, se procede a editar cada uno de estos sectores que hemos grabado y a transferirlos a un fichero de QL.

¿Fácil, verdad? La única pega es que es un rollazo editar los 14 sectores con el SMM. Un truco: Si se dispone del TaskMaster se puede crear un fichero "_cmd" de modo que sólo tengamos que hacer el trabajo una vez.

De lo único que hay que tener cuidado es de reformatear el disco que grabamos con SAVE @ pues de él no se pueden sacar directorios ni nada y queda inservible para cualquier uso "normal".

El sistema es rápido y muy fiable. Una vez que tenemos la pantalla en un fichero con formato QL se puede usar la utilidad que Marcos Cruz publicó en el CUQ número 16 para pasar la pantalla a formato QL. Las pantallas que vienen con este disco fueron todas transferidas por medio de este sistema.

Sergio Montoro Ten
Madrid, 23 febrero 1990

LISP

CAPITULO 3

EL LENGUAJE NATURAL DENTRO DE LA INTELIGENCIA ARTIFICIAL

Introducción y evolución histórica.

Los primeros trabajos sobre lenguajes naturales fueron los relacionados con la traducción mecánica. Durante la década de los 50 surgieron importantes proyectos sobre sistemas de traducción automáticos. Todos se basaban en la creencia de que la traducción era un simple proceso mecánico, que podría ser realizado fácil y rapidamente por los ordenadores. Además, el momento sociológico y político de aquel momento, en el que la "guerra fría" y la falta de comunicación eran constantes en el desarrollo de las relaciones, favoreció el tema, ya que ofrecía el intercambio de ideas e información entre personas de distintos paises sin tener que depender de intérpretes.

Los primeros sistemas desarrollados constaban de:

1.- Un diccionario bilingüe, que relacionaba las palabras de los dos idiomas

entre sí.

2.- Una gramática para cada idioma, que añadía las terminaciones a las palabras y ordenaba los elementos de la oración.

Los escasos avances que se alcanzaron hasta la primera parte de la década de los 60, convencieron a especialistas, tales como Bar Hillel, Weaver y Booth, que la comprensión del lenguaje natural precisaba de algo más que un simple diccionario y una gramática.

Entonces la comprensión del significado de las palabras se basaba en la sintaxis (gramática), sin tener en cuenta la semántica. Los científicos de la época tropezaron con problemas a la hora de utilizar palabras cuyo significado dependía del contexto de la frase, tambien con las expresiones idiomáticas propias y con las frases hechas o las ambigüedades sintácticas.

Por ejemplo, supongamos que se quiere traducir del inglés al castellano la siguiente frase:

Mary attended the ball

La palabra ball significa baile en la frase anterior. Sin embargo, en la expresión:

Mary kicked the ball

La palabra ball aquí ya no significa baile. El significado de ball depende del contexto en que se encuentre. El traductor humano aprecia inmediatamente la diferencia, pero la máquina no.

Otro ejemplo que muestra los problemas que encontraron los primeros investigadores se deriva del estudio de las dos siguientes frases en inglés:

1. The block of flats was built by the workmen.
2. The block of flats was built by the river.

La palabra by tiene dos usos y dos significados.

Muchos expertos que trabajaban en la comprensión del lenguaje natural, abandonaron su empeño ante las dificultades y otros variaron la orientación de sus estudios y experiencias. En vez de intentar realizar la traducción del lenguaje por medio de ordenadores, se emprendió la tarea de hacer que el ordenador comprendiera el lenguaje.

Hoy en día, los estudios que se realizan sobre el tema que nos ocupa se encaminan hacia la intercomunicación entre el hombre y la máquina a través del lenguaje natural. Tambien se llevan a cabo trabajos en los que se emplea el lenguaje natural como un medio de acceder al cerebro humano.

Se ha avanzado bastante en el campo del entendimiento del lenguaje natural por parte de la máquina. El mayor inconveniente sigue siendo la forma de representar el conocimiento del mundo que nos rodea, en el ordenador.

El ser humano posee una serie de conocimientos que ha ido acumulando a lo largo de su vida y cuando interpreta una pregunta, la responderá tanto mejor cuantos más conocimientos tenga sobre el tema.

Los únicos resultados prácticos que se han obtenido hasta ahora, se han basado en la limitación del dominio de la acción. Es más fácil representar el conocimiento de un "pequeño mundo", al centrarse el sistema en un tema único, en lugar de intentar representar todo el conocimiento.

Así como el entendimiento del lenguaje natural es muy difícil, la generación o síntesis del lenguaje hablado es una técnica muy perfeccionada. Existen en el mercado cicuito electrónicos capaces de generar, con bastante perfección, los sonidos humanos (sintetizador de voz).

Una de las razones que han influido en el progreso de la síntesis de voz, ha sido el conocimiento que se tiene de cómo se produce en el hombre. Los elementos que intervienen en la formación del sonido humano y el cometido específico de cada uno de dichos elementos se conoce perfectamente. No sucede lo mismo con la comprensión del lenguaje. Parece increible la facilidad que poseemos para, con sólo oir una simple palabra, conocer el sexo, el estado de ánimo, e incluso, la intención del que la ha pronunciado.

La comprensión de las palabras.

La mayor parte de la comunicación humana se realiza mediante el lenguaje oral. Sin embargo, es mucho más fácil entender el lenguaje escrito que el oral.

Para construir un programa que entendiese el lenguaje oral, habría que tener en consideración, además del mecanismo que fuese capaz de "entender el contexto", la entonación, el acento, las pausas e, incluso, el ruido ambiental.

Existen dos campos en el estudio del lenguaje natural:

1.- Entendimiento del lenguaje escrito, que utiliza el conocimiento léxico, sintáctico y semántico del lenguaje, unido a la información o conocimiento del dominio.
2.- Entendimiento del lenguaje oral, que comprende todo lo del apartado anterior junto con toda la fonología.

Nos ceñiremos en el entendimiento del lenguaje escrito y veremos las técnicas que se pueden emplear para entender oraciones simples y no relacionadas entre sí. Para poder entender un párrafo con varias frases habría que proporcionar a la máquina la habilidad de buscar las relaciones entre la oraciones. Todas las complejas técnicas usadas en la comprensión del lenguaje natural derivan de la Inteligencia Artificial.
Para entender una simple oración, hay que considerar tres aspectos:

a) Comprender el significado de cada palabra de la oración.
b) Comprender la estructura de la oración o frase.
c) Tener el conocimiento del contexto en que se pronuncia.

Para entender el significado de cada palabra, se puede incluir en el sistema un diccionario que contenga a todas ellas junto a lo que quieren decir. Pero hay palabras con varios significados. Por ejemplo, la palabra gato tiene dos significados. ¿Cómo sabría la máquina tomar cuál de ellos corresponde a la siguiente oración?
Empleó el "gato" para levantar el coche.
Para distinguir palabras con diversos significados se podría asociar a cada palabra de este tipo una información adicional sobre el contexto en el que puede aparecer. Pero esto resulta impracticable en muchas ocasiones. A la palabra gato habría que proporcionarle todas las características que puede tener cuando representa un animal (cuatro patas, bigotes, mayar, etc.), que conformarían un paquete de información gigantesco. Además, cada una de estas características podría hacer referencia a otra muchas.

La comprensión de las frases.
Otra de las consideraciones en la comprensión del lenguaje natural escrito, además del entendimiento de las palabras por separado, es "entender la estructura de la oración", que también es difícil. Este tema se puede dividir en tres análisis diferentes:

1.- Análisis sintáctico de la oración.
2.- Análisis semántico.
3.- Análisis interpretativo.

El análisis sintáctico determina la estructura gramatical de la frase. Se efectúa mediante un proceso llamado parsing o análisis de la oración, que consiste en descomponerla en sus elemento constituyentes. Para realizar el análisis de la oración se precisa de una gramática que describa ese lenguaje, para luego construir el árbol de derivación que represente a la sentencia de entrada.
Vamos a ver lo que se entiende por una definición formal de gramática.
Una gramática es una cuádrupla, que viene representada por la siguiente expresión:

G = (VT , VN , P , E)

VT es el alfabeto terminal, es decir, el conjunto de cadenas que forman parte del lenguaje generado por esa gramática.
VN es el alfabeto no terminal, que consiste en el conjunto de "clases gramaticales", que describen el lenguaje generado.
P es el conjunto de reglas de producción o reglas de reescritura.
E es el simbolo de comienzo.
Según sean las reglas de producción, se obtiene un tipo u otro de graática. Se presenta un ejemplo de una posible gramática que genere un reducido grupo de oraciones del lenguaje castellano.

```
<SENTENCIA> ? <SUJETO><PREDICADO>
<SUJETO>    ? <ARTICULO><NOMBRE><ADJETIVO>
<PREDICADO> ? <VERBO><NOMBRE><ADVERBIO>
<ARTICULO>  ? EL
<NOMBRE>    ? RATON
<ADJETIVO>  ? BLANCO
```

```
<VERBO>     ? COME
<NOMBRE>    ? QUESO
<ADVERBIO>  ? AVIDAMENTE
```

Con este conjunto de reglas de producción se obtiene el árbol siguiente, que representa la sentencia:

EL RATON BLANCO COME QUESO AVIDAMENTE

```
                        <SENTENCIA>
                       .           .
                 .                       .
         <SUJETO>                     <PREDICADO>
            .                              .
    .       .       .             .      .       .
<ARTICULO><NOMBRE><ADJETIVO>     <VERBO><NOMBRE><ADVERBIO>
    .       .       .             .      .       .
    .       .       .             .      .       .
    EL    RATON   BLANCO         COME   QUESO  AVIDAMENTE
```

Para cada gramática existe un lenguaje, que es el conjunto de símbolos terminales que se pueden generar a partir del símbolo de comienzo. Ahora bien, siguiendo el ejemplo anterior, se puede formar la frase:

EL QUESO BLANCO COME RATON AVIDAMENTE

Para que la máquina sepa que no puede generar esta sentencia, habría que haberle indicado que el queso no tiene la propiedad de comer.

Para un lenguaje dado (recuérdese que el lenguaje es el conjunto de cadenas generado a partir del símbolo de comienzo aplicando las reglas de producción) puede que exista más de una gramática. Las gramáticas que generan un mismo lenguaje pueden ofrecer problemas a la hora de hacer el análisis. A dichas gramáticas se las califica de "ambiguas" y producen diversos árboles.

El análisis semántico es el segundo paso en el entendimiento de la estructura de la oración. Una vez descompuesta la oración, se procede a la interpretación semántica. Existen varias formas de resolver esta frase:

a) Gramáticas semánticas, son gramáticas libres de contexto, que combinan todo tipo de conocimientos, tanto sintácticos como semánticos en un solo tipo de reglas de producción.

b) Gramáticas basadas en el caso, en las que el análisis aplicado contiene cierta información semántica. Por ejemplo:

MARIA BESO A PEDRO
PEDRO FUE BESADO POR MARIA

Los papeles semánticos de María y Pedro son los mismos, pero los papeles sintácticos están cambiados en las frases.

Si se usa una gramática basada en el caso, la asignación de casos es la misma, aunque ha cambiado la estructura superficial.

La representación será:

(BESO(AGENTE:MARIA)(DATIVO:PEDRO))

Por último, el análisis interpretativo traduce lo que se ha escrito a lo que se quería decir. Por ejemplo en la frase:

¿SABE USTED QUE HORA ES?

La respuesta podría ser SI o NO, pero en realidad eso no es lo que se espera que conteste la máquina. Queremos saber la hora concreta.

Como conclusión, el entendimiento del lenguaje natural es, hoy por hoy, muy difícil por parte de una máquina y casi todos los problemas que presenta están por resolver. Aquí sólo se ha querido dar una idea de algunos problemas y soluciones, pero queda mucho por hacer.

Algunos sistemas de proceso de lenguaje natural construidos.

Los primeros sistemas que se construyeron sólo operaban con la información sintáctica de la sentencia y con el significado de un conjunto de palabras. Se limitaban a diálogos sobre dominios muy reducidos, en idioma inglés básico. Entre los sistemas construidos más conocidos, destacan: SAD-SAM, ELIZA,

BASEBALL, SIT y STUDENT.

El sistema Baseball.

Bert Green escribió este programa, que esencialmente consiste en recuperar información sin modificar la base de datos, que almacena los hechos relevantes de todos los partidos de la liga americana durante un año.

Las preguntas que puede realizar el usuario solo deben tener una cláusula y carecer de conectiva como "y", "o", etc. Tampoco admite comparativos como "más alto", "menos largo" y similares.

Un ejemplo de pregunta al que podría contestar el sistema BASEBALL sería la siguiente:

¿Cuántos partidos jugaron los Yankees en julio?

La respuesta se encuentra buscando en la base de datos los elementos que tengan que ver con las especificaciones de la pregunta, almacenándolos y procesándolos posteriormente.

El programa BASEBALL contesta la mayoría de las preguntas razonables que se hagan sobre los juegos almacenados en la base de datos, dada la gran magnitud del diccionario y la heurística empleada en resolver ambigüedades sintácticas. Cuando no comprende una pregunta, pide al usuario que la construya de forma diferente.

El sistema Eliza.

Este programa fue escrito en 1966 por Joseph Weizanbaum en el MIT y toma el papel de un psicoanalista en un diálogo con el usuario.

Aunque el diálogo (en inglés) parece sorprendentemente real, toda la estructura heurística o inteligente sólo se basa en el emparejamiento de las sentencias con plantillas ó modelos. Seguidamente se muestra un posible diálogo con el ELIZA.

E (ELIZA): Buenos días ¿En que puedo ayudarte?
P (PACIENTE): Todos los hombres son iguales.
E: ¿En qué son iguales?
P: Siempre nos están empujando a hacer una cosa u otra.
E: ¿Puedes darme un ejemplo específico?
P: Bueno, pues mi novio me obligó a venir aquí.
E: ¿Por qué tu novio te obligó a venir aquí?
P: El dice que estoy muy deprimida.
E: Lo siento que estés deprimida.
P: Es verdad. No soy feliz.
E: ¿Crees que venir aquí te ayudará a ser feliz?

El programa ELIZA interpreta las sentencias usando normas que especifican el conjunto de modelos que habrá que buscar para una palabra clave dada y las posibles transformaciones asociadas con cada modelo.

El sistema lunar.

Se trata de un sistema de recuperación de información que tiene la misión de ayudar a los geólogos a acceder, comparar y evaluar los datos de los análisis químicos realizados sobre rocas de la Luna y de la composición de la "tierra", que recogió el Apollo-II. Fue diseñado por William Woods, de forma que las preguntas que se plantean en inglés, se traducen a un lenguaje formal, llamado query. La traducción se hace con un analizador (ATN), unido a un proceso de interpretación semántica que guía el análisis de la pregunta. Lo que resulta del análisis se aplica a la base de datos para generar la respuesta.

Si se hiciese al sistema LUNAR la siguiente pregunta:

¿QUE MUESTRAS CONTIENEN P2O5?

el programa realizaría las tres siguientes fases para poder responderla:

1.- Análisis sintáctico mediante el analizador ATN e información heurística para producir el árbol de derivación más correcto de la pregunta.

2.- Interpretación semántica para producir una representación del significado de la pregunta en query.

3.- Ejecución de la expresión del lenguaje query en la base de datos para producir la respuesta.

El procesador de lenguaje LUNAR contiene una gramática ATN para un gran subconjunto del inglés, reglas semánticas para interpretar preguntas de la base

de datos y un diccionario de unas 3.500 palabras.

El sistema SHRDLU.

SHRDLU es un sistema de procesamiento de lenguaje natural escrito por Terry Winograd en 1972, como trabajo de investigación para su doctorado en el MIT. Lo escribió en los lenguajes LISP y MICRO-PLANNER, derivado del primero.

El programa simula el funcionamiento del brazo de un robot que manipula bloques de juguete sobre una mesa.

El sistema mantiene un diálogo interactivo con el usuario, pudiendo aceptar sentencias, órdenes y, también, contestar preguntas sobre el estado de su mundo y la razón de sus acciones.

El sistema consta de un analizador para un subconjunto simple de construcciones sintácticas, que detecta los constituyentes (grupos nominales, preposicionales, etc.) y a continuación un conjunto de procedimientos (asociados a las palabras) que construyen la interpretación y algunas modificaciones.

Supongamos un robot situado en una habitación donde se encuentran diversos objetos, algunos que se pueden mover (un bloque azul y una pirámide verde) y otros no, como por ej. una mesa y una caja verde con un bloque rojo y una pirámide azul en su interior.

El tiempo se modela con una variable T que se incrementa en una unidad con cada movimiento del brazo del robot. Sobre una situación como la descrita anteriormente, un ejemplo de posible diálogo sería el siguiente:

Pregunta: ¿Qué hay en la caja?
Respuesta: "El cubo rojo, la pirámide azul"
Orden: "Coge el cubo rojo."

Esta acción la efectúa el robot desplazando el brazo a la posición donde está el cubo. Otra orden ante las nuevas coordenadas del objeto sería:

Orden: "Ponlo sobre la mesa."

Con esta última orden se hace referencia a un objeto de la fase anterior, el robot determina el objeto y lleva a cabo el desplazamiento.

Orden: "Pon la pirámide verde encima."

Ahora, el objeto está definido en la orden, pero el lugar de destino es una referencia a la frase anterior.

Orden: "Pon la pirámide azul sobre el bloque más grande."

Primero busca el "bloque más grande" y luego ejecuta la acción.

Pregunta: ¿Cuál es la pirámide que está sobre el cubo que estaba en la caja?

El robot consulta su "memoria" y así produce como respuesta la pirámide verde.

En el caso de hacer preguntas sobre el tiempo en que se produjo algún movimiento, consulta la variable T.

El sistema SHRDLU consta de cuatro elementos básicos:

- Analizador
- Gramática semántica
- Programas para análisis semántico
- Programa de solución de problemas en las tareas en el mundo de los bloques.

El analizador cuenta con una memoria de diálogo. Produce un árbol de derivación y un programa en lenguaje PLANNER, que corresponde a la pregunta u orden dada al robot.

El diseño del sistema se basó en la creencia de que, para comprender el lenguaje, un programa debe tratar de una forma integral la sintaxis, la semántica y el razonamiento.

El sistema Lifer.

Fue creado por Gary Hendrix en 1977 y sirve para construir los principales objetivos de un lenguaje natural para su aplicación en cualquier dominio concreto. Lo usan los diseñadores de sistemas para crear un programa que interprete la entrada en inglés y produzca la secuencia apropiada de comandos para su sistema.

La flexibilidad del sistema se manifiesta en que, tanto los diseñadores como sus posibles usuarios, pueden modificarlo de forma que se adapte a las necesidades particulares.

El sistema LIFER se compone de dos partes fundamentales:
1.- Un conjunto de funciones interactivas para especificar un lenguaje.
2.- Un analizador.

Inicialmente no contiene gramática ni semántica para algún dominio del lenguaje. Un constructor de interfaz usa las funciones de especificación del lenguaje para definir un lenguaje de aplicación, que será un subconjunto del inglés, apropiado para interactuar con su sistema de aplicación. LIFER usa esta especificación del lenguaje para interpretar las entradas en lenguaje natural como comandos para el sistema de aplicación.

El constructor de interfaz define el lenguaje, principalmente en términos de reglas de reescritura gramaticales. LIFER las traduce automaticamente a árboles de transición, que son una forma simplificada del analizador ATN. Usando dicho árbol, el analizador interpreta las entradas en el lenguaje de aplicación. El resultado es la interpretación, en términos de las rutinas apropiadas del sistema de aplicaciones, tal como hayan sido especificadas por el constructor del interfaz.

LIFER ha tenido diversas aplicaciones. Se ha usado para construir un gran número de interfaces de lenguaje natural, como una base de datos médica o un sistema experto. El sistema más complejo construido con un interfaz LIFER es LADDER, que proporciona un acceso en lenguaje natural y en tiempo real a una gran base de datos esparcida por muchas bases de datos más pequeñas situadas en ordenadores localizados en diversos puntos de EE.UU. Los usuarios no necesitan tener conocimiento alguno sobre la organización de los datos ni sobre su almacenamiento.

Carlos Iglesias del Río. Madrid 1-3-90

¿Sabéis que nuestro ordenador puede ver? Pues...¡Esperad al próximo CUQ!

L-TOOLKIT

Empiezo por pedir disculpas a quienes hayan usado el comando ORDER del L-TOOLKIT y se hayan visto en problemas pero es que un VIRUS llamado HUMANIDAD se coló en mi QL y provocó desmadres con el Toolkit.

Hay que tener cuidado con las rutinas etiquetadas como "subrp", tanto el el manual original como en la traducción de yo hice, porque algunas de ellas no responden correctamente a la función (subrp U:cualquiera). Por ello es necesario eliminar de la función ORDER la linea que testea que la condición de ordenación sea de tipo subrp. Por lo demás, ORDER funciona perfectamente.

Había otro pequeño lío con las funciones EVERY y FIND-IF y la forma en la que aceptaban sus parametros; por ello, en el listado que viene a continuación va una definición de ambas perfectamente acorde a las instrucciones de L-TOOLKIT.

```
(DEFUN EVERY (pred list (status))
  (SETQ status t)
  (LOOP (UNTIL (NULL list))
     (SETQ status (AND status (pred (CAR list) ) ) )
     (SETQ list (CDR list)) ) status )

(DEFUN FIND-IF (pred list (status))
  (SETQ status nil)
  (LOOP (UNTIL (OR (NULL list) (NOT (NULL status))))
     (SETQ status (OR status (pred (CAR list))))
     (COND (status (SETQ status (CAR list))))
     (SETQ list (CDR list)) ) status)

(DEFUN ORDER (l order-cond (reps count local))
  (COND ( (NOT (LISTP l))
          (ERROR '(attemp to order a non-list argument)) ) )
  (SETQ reps (LENGTH l))
  (REPEAT reps
     (SETQ count 1)
     (LOOP (UNTIL (EQ count reps))
        (COND ((order-cond
              (NTH count l)
              (HEAD count l) )
           (SETQ local (PRELIST count l))
           (SETQ local (APPEND local (CONS (NTH count l) nil)))
           (SETQ local (APPEND local (CONS (HEAD count l) nil)))
```

```
            (SETQ local (APPEND local (CDR (NTHCDR count l)))) )
            ( T
            (SETQ local (PRELIST count l))
            (SETQ local (APPEND local (CONS (HEAD count l) nil)))
            (SETQ local (APPEND local (CONS (NTH count l) nil)))
            (SETQ local (APPEND local (CDR (NTHCDR count l))))))
      (SETQ l local)
      (SETQ count (ADD1 count)) )
      (SETQ reps (SUB1 reps)) )
      l )
```

Sergio Montoro Ten
Madrid, 23 febrero 1990

LENGUAJE SuperBASIC

FERRARI

Sólo quiero dar las gracias al autor desconocido de este programita. Hasta mis manos llegó sin © ni nombre del autor, así que creo que no le importará que el coche pase a los usuarios del CUQ.

Sergio Montoro Ten
Madrid, 23 de febrero 1990

PROGRAMA VERB_TSK

Los verbos con sus distintas formas desarrollados se almacenan en el drive II. Al introducir un nuevo infinitivo el programa comprueba en el drive II si no está ya hecho, y en caso de que sea asi se crea un fichero con su nombre (_dic) en el que incluye las formas. Lo he hecho asi para no tener que andar haciendo listados previos.

Luego pueden agruparse, importándoloes a un tratamiento de textos para revisión de formas en el caso de introducir irregulares y posterior incorporación al diccionario. Para salir del programa se usa 00 (cero cero).

Dasio Carballeira Tella
Santiago de Compostella, 25/2/1990

Z88

Z88 FORTH

En los últimos CUQs, os he estado poniendo al dia de un proyecto que más de uno habrá pensado que era imposible. Pero la historia para engordarla un poco hay que empezarla por el principio.

Todo empieza cuando recibo una carta de Borja Marcos con la intención de convencer a todos los socios de FIG en España para crear un grupo local. En su carta anunciaba su proyecto de escribir un FORTH para el Psion Organiser, y eso fue lo que me dió la idea de escribir mi propio FORTH para el Z88.

Conseguir toda la documentación necesaria fue bastante fácil por mediación de FIG California, pero costó en dólares tanto como comprarlo ya hecho.

Pero ¿Porqué escribir un FORTH para el Z88?

El uso principal que queria darle al Z88 era tomar datos o informes en el lugar de origen, e incluso escribir programas para el QL en cualquier lugar. El BBC BASIC no me interesaba en absoluto y programar en assembler Z80 no me hace mucha gracia. Aunque hace unos 6 años era programador Z80 traduciendo a mano sin la ayuda de un ensamblador, pero eso es otra historia (¿Adivinad por qué me gusta el QL?). Sin embargo, instalar el FORTH era una buena idea, pues es un

lenguaje que me gusta.
A continuación vais a ver un listado Plain TEXT que se carga en el BBC BASIC con el comando *CLI .*forth.cli

```
.J
NEW
10 DIM code 50
20 DIM buff 25
30 DIM filename 40
45 DIM fth 38900
50 $filename=":ram.1/forth.cde "
120 FOR pass=0 TO 2 STEP 2
130 P%=code
140 [OPT pass
150 LD HL,0
160 ADD HL,SP
170 LD SP,(&1FFE)
180 PUSH HL
190 CALL main
1100 POP HL
1110 LD SP,HL
1120 RET
1130 \
1140 .main
1150 LD HL,filename
1160 LD DE,buff
1170 LD BC,20
1180 LD A,1
1190 RST &20:DEFW &6009
1200 LD BC,4037
1210 LD DE,10000
1220 LD HL,0
1230 RST &20 :DEFB &45
1240 RST &20:DEFB &09:DEFB &62
1250 LD A,6
1260 RST &20 : DEFB &6F
1290 RET
1300 ]
1310 NEXT pass
1320 CALL code
1330 CALL 10000
1340 END
```

El anterior programita escrito en BBC BASIC lo único que hace es reservar 38900 bytes para el sistema FORTH, cargar en la dirección 10000 el fichero FORTH.CDE, desactivar el BREAK (pues si se pulsa la tecla ESC el programa intenta volver al BASIC y cuelga todo el sistema) e inicia el sistema FORTH.
La verdad es que los diseñadores de sistemas operativos y lenguajes como el BBC BASIC deben tratarse de programadores con "mala leche". El equivalente en Superbasic del cargador anterior, seria:

```
10 a=RESPR(38900)
20 LBYTES flp1_forth_cde, 10000
30 CALL 10000
```

Para colmo, en el manual Cambridge Z88 Developer's Notes dice que programas como este Z88 FORTH son aplicaciones mal escritas. Una aplicación bien escrita es aquella que ha sido escrita y grabada en una EPROM con su correspondiente cabecera y otros extras. Lo que pasa es que las EPROM son muy caras, grabar y borrar es muy laborioso (20 minutos tarda un borrador de Eproms), y además, a mi la memoria dividida en segmentos de 16 Kbytes usando un byte para el número de banco y dos bytes para la dirección dentro del banco, y gestionando la memoria no más de 256 bytes continuos, no me gusta ni me apetece aprender a usar.
Z88 FORTH ha sido escrito totalmente y ensamblado en el QL con la ayuda del Quantum Z80 (de Marcos Cruz). Aunque la versión que he tenido el honor de usar era una auténtica bestia negra peluda llena de errores, los cuales he tenido que descubrir y acostumbrarme a ellos para poder cumplir mi objetivo, el Z88 FORTH V$ 1.00.
Z88 FORTH no es Standard, pero puedo convertirlo a algún standard si lo desease. Un ejemplo podria ser:

```
: HOLA [ ¡HOLA, SOY Z88 FORTH!] CRET ;
: DIME [ ESCRIBE UN NUMERO ] NUMBER CRET ;
: SOLUCION [ EL CUADRADO DE ]
             IF 2DUP . [ ES ] * .
               ELSE [ .... ¡QUE PASA CON ESE NUMERO!]
                                                    THEN ;
: PROGRAMA CLEAR HOLA DIME SOLUCION ;

PROGRAMA
```

Para finalizar el presente articulo o carta, según deseemos bautizarlo, os diré que aunque Z88 FORTH V$ 1.00 tiene más de 150 palabras en su diccionario base, no es la versión definitiva. La próxima versión va a disponer de las siguientes extensiones:

- Z80 Forth Assembler.
- Paquete matemáticas en coma flotante.
- Manejo de ficheros y otros dispositivos.
- Gráficos (256x64 pixels).
- Opción Stand-Alone.
- Y acceso a muchas llamadas OZ.

Salvador Merino, 2/3/1990

OFERTAS

COMPRO revistas o fotocopias de revistas QL atrasadas. Carlos Mestre, Calle José Antonio 82, 12.580 Benicarlo (Castellón).

INTERCAMBIO programas QL comerciales por coleccionismo. Enviad lista a Salvador Merino, Ctra Cádiz, Cerámicas Mary, 29640 Torreblanca del Sol (Málaga).

USUARIOS Z88, si no tienes programas, ponte en contacto conmigo. Salvador Merino, Ctra Cádiz, Cerámicas Mary, 29640 Torreblanca del Sol (Málaga).